NudtMDP at TREC 2015 LiveQA Track

Yuanping Nie, Jiuming Huang, Zongsheng Xie, Hai Li, Pengfei Zhang and Yan Jia

College of Computer, National University of Defense Technology, China
{Yuanpingnie,Zongshengxie,Pengfeizhang,HaiLi}@nudt.edu.cn
Jiuming.Huang@qq.com

Abstract. In this paper, we describe a web-based online question answering system which has been evaluated in TREC 2015 Live QA task. Automatic question answering is a classic widely learned technology. TREC have host 8 times QA tracks since 1999. However, the TREC results show that there is still a long way to solve the questions perfectly. LiveQA is kind of questions means asked by 'real users'. Most liveQAs are non-factoid questions and it is much more challenge to answer the liveQAs than factoid QAs. We build a question answering system to find the answers from web data. The system has two channels, one use search engine to obtain the answers and the other focus on community question answering websites. We finally submit 3 runs in the official test. Two of our runs can perform much better than the avarge scores.

Keywords: web based search, question answering, CQA

1 Introduction

In this paper, we describe the online Question Answering system which has been evaluated in TREC 2015 Live QA task. In this year Live QA track, there is only one main task, which has aimed at the task of providing automatic answers for human nature questions. The live QA track, different from past QA tracks is focusing on 'live' questions that are from real-user. All the testing questions are from Yahoo Answer. YA questions have many question types such as opinion, advice, polls, which make the task much more difficult. The answers also have length limitation, the answer length restriction is 250 chars at first and then increased to 1000 chars. We think the limitation of answers' length is another challenge for this year QA task. We notice that the testing questions are quite difficult. Firstly there is no given source for answers. We need to find answers through Internet and the answers must be extracted as a short query(1000chars). Secondly, all of the questions are from Yahoo Answer questioners. Most questions are asked by spoken language. There exist many oral words and the style of sentences is also complicated. Thirdly the knowledge and information that questioners required are subjective and divergent. In this year Live QA track, lots of questions may have many available answers. And some questions need professional answers. So we decide to crawl and extract the answers by using search engine(Google) and CQAs(Community Question Answering websites). This is our first participation in TREC. Our primary goal this year is to develop a Question Answering system framework to which future enhancements can be applied. We finally submit three runs for the official run in Sept.2. One system(nudtmdp 2) just use CQAs which include eight community question answer websites such as Yahoo Answer, AnswerBag and Answers as answers resource. The other two systems(nudtmdp 1 and 3) employ both CQAs and search engine as answers resource. The difference between this two systems is just the different strategy of choosing answers, which we will introduce later. The results show that the two fusion version QA systems(0.670 and 0.602) performed much better than the CQA version(0.388) and the averge score(0.465).

In this paper, we introduce the main idea and framework of our QA system. The structure of the QA system is shown at Fig 1. It contains three parts: Question Processing Part, Distributed Crawler Part and Answer Processing Part. The Question Processing Part is to translate the question to the search query. It has three components, Classification: we classify the questions into different categories; Filter: the stop words and no useful information can be removed automatically. Extention: We use Word2Vec [2] and some dictionary to extend the search query in order to acquire more useful results. After question processing, we employ a distributed crawler to search the results. For we have used search engine(Google) and serval CQA sites as the answers resources, we need a stable distributed crawler for the QA systems. In this year, we use Apache Storm[] framework. Finally the Answer Processing Part contained the strategy to choose the best answer and extract the answerto 1000 chars.

In the following sections, we describe in detail the parts taken to create our runs. We also discuss what we have learned from this exercise.

2 System Overview

In this section, we introduce our question answering system framework for the live QA task. Our Question Answering system can automatic answer user's nature language questions based on online search. The input of the system is a nature language question and the outputs are the best answer selected from candidate answers by the ranking scores. The QA system can response one question within 1 min. Our Question answering system includes several phases. And it can be broadly divided into three main parts: Question Processing Part, Distributed Crawler Part and Answer Processing Part. The structure of our QA system is shown at Fig.1.

2.1 Question Processing Part

The first step of the question answering system is to understand the questioner's nature language questions and translate the questions to search queries. In this year Live QA task, all the testing questions are from Yahoo Answer, which are consisted by two parts: Titles and Bodies. The titles are necessary for every

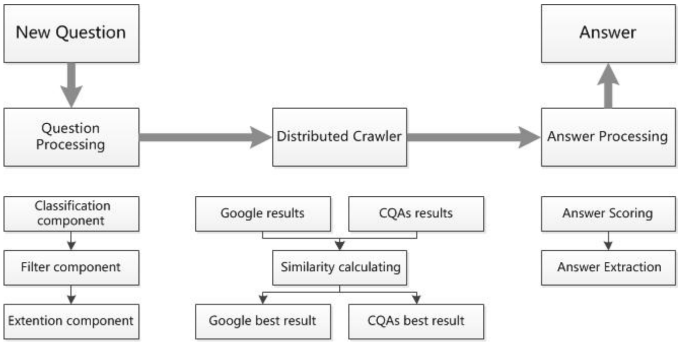

Fig. 1. The structure of our QA system

;

questioners and the bodies are not. We notice that most questions only have titles or the titles already have the whole useful information that meet the questioners' need. The other questions have both titles and bodies. The bodies usually are descriptions for titles and have a much longer length. We found that using the titles as questions are much easier to process. However in this year QA task, we employ a primary strategy to use titles and bodies. The strategy is shown at below:

The Method of using titles and bodies

```
Begin
If title is not empty and Body is empty;
Use title as question;
If title is not empty and Body is not empty;
If title ¿ T;
Use only title as question;
else
use title and abstract of body as question
end
```

And we use such components to process the nature language questions. 1) Classification component: The given questions from Yahoo Answer do have category features, like Pets, History and Hair. However, we find the categories is to large and different community question answering systems have their own classification systems. So it is necessary to rebuild the classification system, which is more detailed by ourself. Based on the exist categories, we employ Latent Dirichlet Allocation(LDA) [1] model, which is a hierarchical nonpara-metric Bayesian approach to discover topic in text corpora training new categories. The classification component can help the system to understand the questioner needs. 2)

Filter component: We remove the stop words and no useful information such as oral words. We also abandon the non-English words by a language detector called Idig [3]. This tool kit is a prototype for short message service with 99.1% accuracy for 17 languages. We only keep the questions which are consisted by the vast majority English characters with a threshold value. The filter component increase the efficiency and accuracy of the question search. 3) Extention component: In the QA system we use Word2Vec, and some dictionary to extend the search query in order to obtain more useful results. Some abbreviation such as WW2 should be expanded to world war two and ASAP to as soon as possible. There are also some proper names can be extended, for example the Summer Palace can be extended to Summer Palace, Beijing, China. For the questions are usually quite short, expending the information of questions can search more useful results.

2.2 Distributed Crwaler Part

In this year Live QA task, we employ Google and CQAs as the answers resources. The official runs have time limitation: the answer can not be reached over 60 seconds. So it makes great demands on our crawler. We use Apache Storm [5] as our crawler framework. Apache Storm is a free and open source distributed realtime computation system. Storm makes it easy to reliably process unbounded streams of data. We use eight famous community question answering systems are our CQA resources, such as: Yahoo Answer, AnswerBag, Askville, About and so on. One of the most important component is to calculate the similarity between target questions and candidate questions. Similarity calculating component: Calculating the similarity between two questions is a very important component in our QA systems. In search engine and community question answering web sites we can always find candidate questions or answers. How to find the most related information is one of the major task we should solve. In the QA system, we jointly consider the words similarity, words dependency and sentence structure to calculate the similarity of two given questions. To normalized the equation, we get the formula of the similarity calculation is shown at Eqn.1; If there are two questions a and b, we can calculate the similarity of two questions as:

$$Sim = A * X * B^T + A^T * X * B \qquad (1)$$

where, X is a similarity matrix betweem question a and b. Every element of X means the distence of every two words of sentence a and b, which can be measured using WordNet [4]. The A and B is a dependency matrix of question a and b. In A every element takes the dependency of every two words in question a. However, we find that it is much more difficult to calculate the similarities between questions and answers. We do some label works which show there is only very weak semantic link between questions and answers. In CQAs there are no such problems, for we should just judge the similarity of two similar questions. But in search engine such as Google, the search results are not questions. In this paper, we use the title or key sentence of the articles as 'questions' so that we can avoid to judge the similarity of questions and answers.

2.3 Distributed Crwaler Part

Answer Process Part is to extract the right answer from the candidate results. According to answer processing part, the similarity component can obtain the best question from the candidate questions we search from Internet. As we know, every similar question in CQAs may has more than one answer. Which answer is the best answer of the most similar question is the first component of the Answer Processing Part. And the next part is to extract the answers meeting the length limitations.

Answer score: In community question answering websites, we did not use the Nature Language Processing(NLP) method to choose the best answer, because there are many user behaviors can be used. We employed the users behavioral analysis to score the candidate answers. The behavioral information includes: a) Many CQA systems allow questioner choose the best answer(or other similar name) by themselves. There some exist questions have the best answer already. b) The websites like Yahoo Answer allow other users give every answer their attitude such as support(up) and argue(down). c) Answerers have their reputation in CQA systems. We believe that a good answerer can provide good answers. So we employed the answers reputation in our QA system. The reputation is calculated by their question-answer history and the relevancy of the topic.

Answer Extraction: When the answer length limitation is 250 chars, the answer extraction is very important to the candidate results both from CQAs and Google. We can only get 1-2 sentences for the answers. In this situation, the CQAs' results perform better, because the answers of CQA are much shorter than search engine results. When the answer length limitation is extend to 1000 chars, the situation is different. There is little influence in CQAs' results, but in the Google results, they are much easier to gain good answers. In this paper, we focus on introducing our Google result's extraction strategy. We first open the website and find the paragraph that contains the matching information. Then we calculate the Importance of the first sentence and the relevancy to next sentence and go through. Then we can get the score of our sentences. We choose as many as possible sentences to consist our answers.

3 Evaluation

Evaluation method: We submitted three runs for main task at Live QA in Trec 2015. There were 1087 questions judged and scored using 4-level scale:

4: Excellent a significant amount of useful information, fully answers the question

3: Good C partially answers the question

2: Fair marginally useful information

1: Bad C contains no useful information for the question

-2: the answer is unreadable (only 15 answers from all runs were judged as unreadable)

The performance measures are:

avg-score(0-3) : average score over all queries (transfer- ring 1-4 level scores to 0-3, hence comparing 1-level score with no-answer score, also considering -2-level score as 0)

succ@i+ : number of questions with i+ score (i=1..4) divided by number of all questions

prec@i+ : number of questions with i+ score (i=2..4) divided by number of answered only questions Our three runs results and the average scores of all runs are shown below.

Run Name	Avg Score	succ@1+	succ@2+	succ@3+	succ4+	prec2+	prec3+	prec4+
NUDTMDP1	0.670	0.958	0.353	0.210	0.107	0.369	0.219	0.111
NUDTMDP2	0.388	0.942	0.228	0.120	0.043	0.242	0.127	0.046
NUDTMDP3	0.602	0.952	0.319	0.186	0.097	0.355	0.195	0.101
Average	0.465	0.925	0.262	0.146	0.060	0.284	0.159	0.065

Table 1. The results of our 3 runs and the average result

4 Conclusion and Discussion

This is our first time participation in LiveQA task. Our primary goal this year is to build an automatic question answering system that can be applied in next years tasks. In the beginning, we firstly try to use knowledge based question answering system to solve the questions. We have built a question answering system using Wiki, Freebase and DBpedia as knowledge base. We crawl some history Yahoo Answer QA pairs as testing database. However, the results of our system are not satisfied. We think the reason may be: the knowledge based answering system can solve objective questions which are contained in the database with high performance. But if the questions knowledge doesnt contain in the database, the system can not give good answers. In this year liveQA task, all the questions are from real person and most of them are subjective. So the performance of our knowledge based QA system is not good. Then we employ search engine based QA system. This QA system version has one important advantage, it is that there are always results when search the search engine. There are also some challenges in search engine based QA system, the results we obtained are very long. And the limitation of answers is no larger than 250 chars at first. It is pretty difficult to extract the accurate information from candidate results. In recent years, community question answering(CQA) systems become very popular. CQAs have many exist question-answer pairs. We find use CQAs based question answering system can solve many questions. However, there still have some questions can not be good answered. So we finally decide to use CQAs and Google as our question answering systems resources. However, we still have many parts which have not been good solved. Some exist components are just using primary strategy. In next year LiveQA task, we will focus on those components and improve our approaches.

Acknowledgement

This work was supported in part by the National Key Fundamental Research and Development Program of China (2013CB329601), National Natural Science Foundation of China (61372191, 61202362, 61472433), Project funded by China Postdoctoral Science Foundation(2013M5452560,2015T81129). We are very thankful to Chao An, Hongmei Liu and Yue Qi.

References

1. David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
2. Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.
3. Chao Lv, Feifan Fan, Runwei Qiang, Yue Fei, and Jianwu Yang. Pkuicst at trec 2014 microblog track: Feature extraction for effective microblog search and adaptive clustering algorithms for ttg. Technical report, DTIC Document, 2014.
4. George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
5. Ankit Toshniwal, Siddarth Taneja, Amit Shukla, Karthik Ramasamy, Jignesh M Patel, Sanjeev Kulkarni, Jason Jackson, Krishna Gade, Maosong Fu, Jake Donham, et al. Storm@ twitter. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 147–156. ACM, 2014.